NEURAL SIMULATION OF WATER SYSTEMS FOR EFFICIENT STATE ESTIMATION

B. Gabrys and A. Bargiela
Real Time Telemetry Group
Department of Computing, The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
Tel: +44 (0) 115 9418418 ext 2162, Fax: +44 (0) 115 9486518
E-mail: andre@doc.ntu.ac.uk, bcz@doc.ntu.ac.uk

ABSTRACT

This paper presents a neural network based technique for the solution of a water system state estimation problem.The technique combines a neural linear equations solver with a Newton-Raphson iterations to obtain a solution to an overdetermined set of nonlinear equations.

The algorithm has been applied to a realistic 34-node water network. By changing the values of neural network parameters both the least squares (LS) and least absolute values (LAV) estimates have been obtained and assessed with respect to their sensitivity to measurement errors.

INTRODUCTION

Efficient control of a complex water distribution system requires accurate information about its current operating state. At present in the water industry, modern telemetry hardware systems are being installed to meet these needs. Unfortunately, due to financial constraints, it is not practical to measure all variables of interest. Therefore, the information supplied by the telemetry system must be supplemented by the less accurate predictions of consumptions at the nodes in the network. These predictions are frequently referred to as pseudomeasurements. Measurements and pseudomeasurements are used to calculate flows and pressures in the distribution network through the use of a state estimators which provide a means of reconciling the discrepancies between the mathematical model of the system and the input data (Sterling and Bargiela 1984; Hainsworth 1988). Over the last decade state estimators gradually became the key utility for the implementation of monitoring and control of large scale public utility systems such as water, gas or electric power distribution systems.

With the increasing complexity of modern water distribution systems there is a need for efficient state estimators which will form a basis for the implementation of real time control of these systems. Among the potential algorithms and techniques for state estimation neural network based estimators are of great interest because of their massively parallel nature and consequent computational efficiency. While the full potential of neural networks for mathematical optimization can only be realised with appropriate computing hardware, their performance has been assessed through the simulation studies.

Neuron-like architectures were simulated using MATLAB and SIMULINK programs.The networks have been configured to produce state estimates of the water system according to the least squares and the least absolute values optimality criterias. The results indicate that when implemented in VLSI technology (Cichocki and Unbehauen 1992a, 1992b), the computation time of state estimation task will not be influenced by the size of the problem (assuming the match of the network and the problem size) and the final solution will be found within the time of order of a hundred nanoseconds.

WATER SYSTEM MODEL AND ESTIMATION METHOD

The process of state estimation requires a mathematical model of a water distribution network. The nonlinear head-flow functions describing network elements are used to express mass-balances in each node of the physical system, and to represent the specific measurements that are being taken. This can be expressed as follows:

$$\mathbf{z}=g(\mathbf{x})+\omega \tag{1}$$

where:

$\mathbf{z}$ - measurement vector

$g(\mathbf{x})$ - nonlinear functions describing system

ω - vector of measurement inconsistency

The state estimation can be expressed as a problem of minimization of discrepancies between the actual measurements and the values calculated from the mathematical model.

Using the least squares criterion the state estimation problem can be expressed as:

$$\min_{\mathbf{x}} \; E_2(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{z}-\boldsymbol{g}(\boldsymbol{x}))^T \boldsymbol{W}(\boldsymbol{z}-\boldsymbol{g}(\boldsymbol{x})) \tag{2}$$

Similarly, using the least absolute values criterion the state estimation is expressed as:

$$\min_{\mathbf{x}} \; E_1(\boldsymbol{x}) = \boldsymbol{w}^T|\boldsymbol{z}-\boldsymbol{g}(\boldsymbol{x})| \tag{3}$$

where:

$\boldsymbol{w} \in \boldsymbol{R}^m$ - measurement weight vector

$\boldsymbol{W} = diag[w_1, w_2, \ldots, w_m]$ - measurement weight matrix

While the least squares method (Cichocki and Unbehauen 1992a; Golub 1986; Gill at al. 1981) is optimal for a Gaussian distribution of the noise, the assumption that the set of measurements or observations has a Gaussian error distribution is frequently unrealistic due to various sources of errors such as instrument errors, modelling errors, sampling errors and human errors. In order to reduce the influence of large errors, the least absolute values criterion (3) can usefully be adopted (Sterling and Bargiela 1984; Cichocki and Unbehauen 1992b).

The proposed solution of the state estimation problem (2) or (3) is based on the Newton method. Expanding $\mathbf{g}(\mathbf{x})$ by an initial guess of the state vector $\boldsymbol{x}^{(0)}$, using a first-order Taylor series and

defining $\boldsymbol{z}^{(0)} = \boldsymbol{g}\left(\boldsymbol{x}^{(0)}\right)$, we obtain

$$\boldsymbol{z} = \boldsymbol{z}^{(0)} + \Delta\boldsymbol{z} \tag{4}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}\left(\boldsymbol{x}^{(0)}\right) + \boldsymbol{J}\left(\boldsymbol{x}^{(0)}\right)\Delta\boldsymbol{x} \tag{5}$$

After this linearisation we obtain the following set of equations:

$$\boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} = \boldsymbol{z} - \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \tag{6}$$

where:

$\boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right) \in \boldsymbol{R}^{mxn}$ - Jacobian matrix evaluated at $\boldsymbol{x}^{(k)}$

k=0,1,... - step of the estimation process

Equations (2) and (3) can be therefore expressed as

$$\min_{\Delta\boldsymbol{x}} E_2(\Delta\boldsymbol{x}) = \frac{1}{2}\left(\Delta\boldsymbol{z} - \boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x}\right)^T \boldsymbol{W}\left(\Delta\boldsymbol{z} - \boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x}\right) \tag{7}$$

and

$$\min_{\Delta\boldsymbol{x}} E_1(\Delta\boldsymbol{x}) = \boldsymbol{w}^T\left|\Delta\boldsymbol{z} - \boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x}\right| \tag{8}$$

These overdetermined sets of linear equations form the basis for the construction of a neural network which is presented in the following section.

Since the measurement equations (1) are nonlinear, the solution to (2) or (3) is an iterative process with the consecutive state estimates calculated by under-relaxation of the linear solution

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \gamma\Delta\boldsymbol{x}^{(k)}, \quad \text{k=0,1,...} \tag{9}$$

If all elements of $\Delta\boldsymbol{x}$ in k-th iteration are lower or equal to a predefined convergence accuracy, the iteration procedure stops. Otherwise, a new correction vector is calculated using equation (6) with $\boldsymbol{x}^{(k+1)}$ instead of $\boldsymbol{x}^{(k)}$ and suitable neural network.

NEURAL NETWORKS SOLVING SETS OF LINEAR EQUATIONS

The minimisation problems described by (7) and (8) can be generalised as follows:

$$\min_{\Delta\boldsymbol{x}} E(\Delta\boldsymbol{x}) = \sum_{i=1}^{m} \sigma_i[r_i(\Delta\boldsymbol{x})] \tag{10}$$

where:

E is a general cost (energy) function

$\boldsymbol{A} = \boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)$, $\boldsymbol{b} = \boldsymbol{z} - \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right)$

$r_i(\Delta\boldsymbol{x}) = \boldsymbol{a}_i^T\Delta\boldsymbol{x} - b_i$ is the vector of residuals

$\sigma_i[r_i]$ represents a suitably chosen convex functions.

In a special case when $\sigma_i(r_i) = r_i^2/2$ we obtain the standard least-squares criterion (7) and for $\sigma_i(r_i) = |r_i|$ we have the least absolute values criterion (8).

The minimization of the energy function described by (10) by standard gradient descent methods leads to the following system of nonlinear differential equations:

$$\frac{d\Delta x_j}{dt} = -\mu_j \sum_{i=1}^{m} a_{ij}\left(f_i\left(\sum_{k=1}^{n} a_{ik}\Delta x_k - b_i\right)\right) \tag{11}$$

or in compact matrix form

$$\frac{d\Delta\boldsymbol{x}}{dt} = -\mu\boldsymbol{A}^T f(\boldsymbol{r}(\Delta\boldsymbol{x})) \tag{12}$$

where:

$\mu_j > 0$ is the learning parameter;

$f_i(r_i(\Delta\boldsymbol{x})) = \dfrac{\partial\sigma_i(r_i)}{\partial r_i}$ - activation function

a) for $\sigma_i(r_i) = r_i^2/2$ we have $f_i(r_i(\Delta\boldsymbol{x})) = r_i(\Delta\boldsymbol{x})$ - linear activation function

b) for $\sigma_i(r_i) = |r_i|$ we have $f_i(r_i(\Delta\boldsymbol{x})) = sign[r_i(\Delta\boldsymbol{x})]$ - signum activation function

To cater for both least squares and least absolute values criteria the logistic function can be defined as follows:

$$\sigma_L(r_i) = \frac{\beta}{\alpha} ln(\cosh(\alpha r_i)) \tag{13}$$

where $\alpha > 0, \beta > 0$ are the problem dependant parameters and

$$f_i(r_i) = \frac{\partial\sigma_L(r_i)}{\partial r_i} = \frac{\partial\left(\frac{\beta}{\alpha} ln(\cosh(\alpha r_i))\right)}{r_i} = \beta\tanh(\alpha r_i) \tag{14}$$

is the sigmoid activation function.

Choosing β large with $\alpha = 1/\beta$ the sigmoid activation function approximates $f_i(r_i(\Delta\boldsymbol{x})) = r_i(\Delta\boldsymbol{x})$ over a wide range, so the network solves the system of linear equations in the least squares sense. On the other hand, taking $\beta = 1$ and α large we obtain an approximation of $f_i(r_i(\Delta\boldsymbol{x})) = sign[r_i(\Delta\boldsymbol{x})]$, so the network solves the least absolute values problem.

The system of differential equations (12) has been implemented using SIMULINK, by the artificial neural network (ANN) shown in Figure 1.

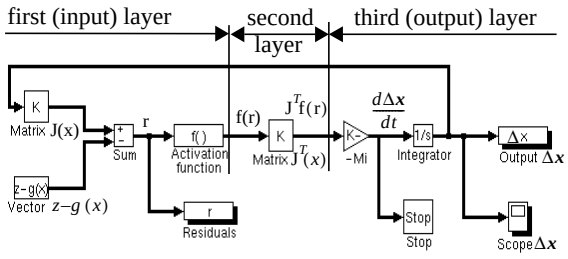

*Figure 1: ANN for solving a system of linear equations (6) based on the system of differential equations (12) with optional activation functions f(**r**) (implementation in MATLAB and SIMULINK)*

COMPUTER SIMULATION RESULTS

The performance of the proposed methods for water-system state estimation was tested on the realistic 34-node network (42 state variables) depicted at the Figure 2.

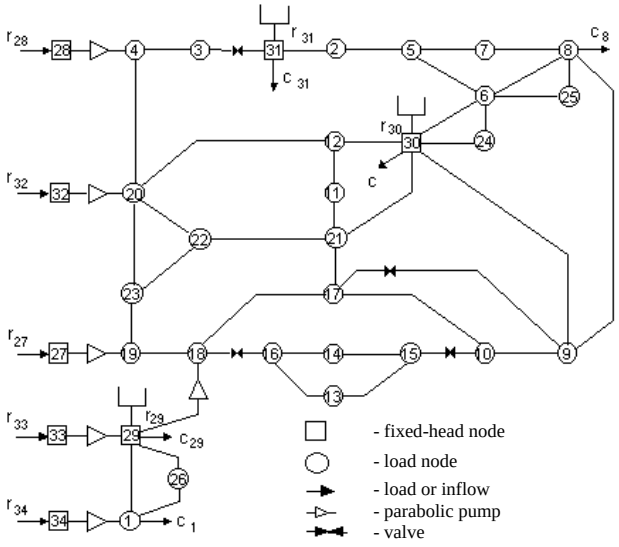

Figure 2: 34-node water distribution network

A complete definition of network parameters are contained in (Sterling and Bargiela 1984). In order to achieve sufficient measurements redundancy (defined as a ratio of the number of measurements and pseudomeasurements to the number of state variables), the set of the mass balance equations was augmented by a number of several flow and pressure measurements.

Two sets of measurements were processed having redundancy ratios 1.74 and 1.4.

The effect of 'bad data' measurements was simulated by introduction of systematic gross errors in head and flow measurements.

The specification of these errors are given in the following examples.

Example 1:

Introduced gross errors

head in node 22 = 42.59 [m Aq] (exact value = 46.59 [m Aq])

load in node 8 = -0.025 [m^3/s] (exact value = -0.075 [m^3/s])

Example 2:

Introduced gross errors:

head in node 22 = 42.59 [m Aq] (exact value = 46.59 [m Aq])

head in node 29 = 35.70 [m Aq] (exact value = 31.70 [m Aq])

head in node 30 = 48.58 [m Aq] (exact value = 43.58 [m Aq])

load in node 8 = -0.025 [m^3/s] (exact value = -0.075 [m^3/s])

Table 1 and Table 2 show the state estimates calculated for redundancy ratios 1.74 and 1.4 respectively. The neural network (Figure 1) has been implemented in SIMULINK. The state vector shown in column 2 of Table 1 and Table 2 is the state vector obtained by exact network simulation and is referred to as a vector of reference values. State estimates of LS and LAV methods calculated for data not including gross errors are presented in column 3 and 4 of Table 1 and Table 2 respectively. State estimates calculated for data including gross errors are presented in columns 5, 6, 7 and 8.

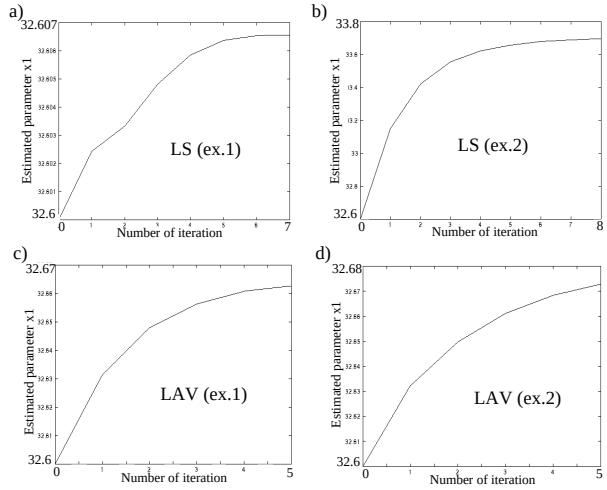

*Figure 3: Estimates of the head in the node 1 (**x1**) using: a) LS estimator for example 1, b) LS estimator for example 2, c) LAV estimator for example 1, d) LAV estimator for example 2*

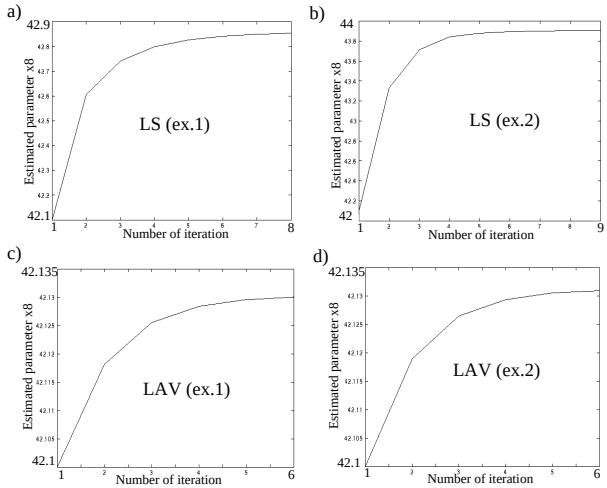

*Figure 4: Estimates of the head in the node 8 (**x8**): a) LS estimator for example 1, b) LS estimator for example 2, c) LAV estimator for example 1, d) LAV estimator for example 2*

LAV (Least Absolute Values) method.

Table 1 (columns 6 and 8) shows the results of examples in which higher measurement redundancy has been used. Table 2 (columns 6 and 8) shows the corresponding results for lower measurement redundancy. Comparison of these results indicates that a smaller number of equations (measurements) was sufficient for accurate estimation with a specific pattern of measurements considered. However, an increased number of measurements contributes mainly to an improved reliability of the estimation and ensures the rejection of a larger spectrum of errors. In conclusion, the LAV problem solution is median solution and passes through at least n (n - number of state variables) of the m data points (measurements). The feature of producing interpolatory fits that approximate closely most of the data while neglecting gross errors is an extremely useful property of the LAV criterion. Provided sufficient basic measurements are available, the LAV estimator

can then act as filter for incoming data

LS (Least Squares) method.

The ordinary LS problem solution is the mean solution since it tries to satisfy all the equations in the set, but usually this solution will not solve exactly any of these equations. The results shown in columns 5 and 7 of Table 1 and Table 2 are very good example of the influence of gross errors on a LS state estimation. A measurement containing gross error has the biggest effect on estimation of the state variables in the node where the error occurred and nodes of the closest vicinity. An increased number of measurements, in this case, helps to reduce an influence of gross errors (averaging process) but the main cause of using of state estimation methods is insufficient number of measurements.

Figures 3 and 4 illustrate the convergence of the estimation process (variables x1 and x8 respectively) for the LS and LAV estimators.

All simulations have been carried out on Sun Workstation using SIMULINK (Dynamic System Simulation Software) and MATLAB (High-Performance Numeric Computation and Visualization Software) programs. Various integration algorithms have been used for many different values of the parameters γ, α, β and μ. The results presented in Tables have been obtained for γ=0.6, $\mu = 1e^6$, Gear integration algorithm. Parameters α and β have been set as follows: α=0.1, β=10 for LS and α=500, β=1 for LAV estimators.

CONCLUSIONS

It has been found, through the simulation study, that a neural network based state estimator provides an efficient means of water system state estimation. While the LS estimates have shown to be strongly affected by any change in the measurement vector, the LAV estimates proved to be resistant to large changes in the data. This is a very useful property when the known data in the measurement vector are contaminated with occasional gross errors.

The main restriction of a VLSI implementation of neural networks is the number of connections between the processing units on a chip. It is envisaged that the future use of optical cross-connecting will enable the implementation of arbitrary large ANN. Consequently the state estimation process, as discussed in this paper, will be accomplished in a time of order of hundred microseconds.

REFERENCES

Cichocki, A. and R.Unbehauen. 1992 “Neural networks for solving systems of linear equations and related problems”, *IEEE Trans. Circuits Syst.*, vol.39 (Feb.), pp.124-138.

Cichocki, A. and R.Unbehauen. 1992. “Neural networks for solving systems of linear equations - Part II: Minimax and absolute value problem”, *IEEE Trans. Circuits Syst.*, vol. 39 (September), pp.619-633.

Gill, P.E.; W.Murray; and M.H.Wright. 1981. *Practical Optimization.* New York: Academic.

Golub, G.H. and C.F.Van Loan. 1986. *Matrix Computations*, Oxford: North Oxford Academic.

Hainsworth, G.D. 1988. “Measurement uncertainty in water distribution telemetry systems”, PhD Thesis, Trent Polytechnic, Nottingham, U.K.

Sterling, M.J.H and A.Bargiela. 1984. “Minimum norm state estimation for computer control of water distribution systems’, *IEE Proceedings*, vol.131 (March), no.2.

State variable	Exact value	LS	LAV	LS (Ex.1)	LAV (Ex.1)	LS (Ex.2)	LAV (Ex.2)
1	32.638	32.6580	32.6607	32.6618	32.6607	33.6298	32.6661
2	43.749	43.7556	43.7586	43.4601	43.7571	43.5068	43.7574
3	46.041	46.0580	46.0421	45.5469	46.0419	45.6997	46.0418
4	46.618	46.6451	46.6222	46.1329	46.6221	46.3419	46.6220
5	43.265	43.2670	43.2713	43.1604	43.2692	43.3690	43.2701
6	43.024	42.9796	42.9711	43.1289	42.9876	43.6542	42.9901
7	42.402	42.3845	42.3708	42.8600	42.4004	43.2596	42.4013
8	42.130	42.1085	42.0966	42.8714	42.1397	43.3611	42.1418
9	43.798	43.7891	43.7830	43.6041	43.7805	44.1570	43.7812
10	47.950	47.9429	47.9476	47.8793	47.9474	47.9314	47.9477
11	44.664	44.6766	44.6705	44.3286	44.6695	44.7447	44.6702
12	44.004	44.0203	44.0145	43.7625	44.0006	44.2883	44.0029
13	49.274	49.2979	49.3000	49.2063	49.3001	49.4836	49.3008
14	49.099	49.0951	49.0977	49.0858	49.0980	49.1191	49.0980
15	49.057	49.0520	49.0536	49.0467	49.0539	49.0665	49.0536
16	49.298	49.3195	49.3212	49.2117	49.3210	49.5452	49.3224
17	47.970	47.9651	47.9684	47.8777	47.9679	48.0098	47.9686
18	49.338	49.3412	49.3423	49.2183	49.3419	49.5987	49.3436
19	49.029	49.0402	49.0379	48.8177	49.0370	49.1068	49.0381
20	46,618	46.6451	46.6221	46.1350	46.6219	46.3518	46.6218
21	45.623	45.6450	45.6313	45.2495	45.6292	45.5800	45.6300
22	46.588	46.6137	46.5920	46.1309	46.5890	46.3555	46.5891
23	48.379	48.3864	48.3798	48.0869	48.3784	48.3482	48.3792
24	43.249	43.2313	43.2244	43.1863	43.2334	43.7602	43.2362
25	42.532	42.4993	42.4885	42.9640	42.5194	43.4724	42.5217
26	32.086	32.0980	32.1013	32.0906	32.1013	33.1495	32.1066
27	-15.233	-15.1963	-15.1962	-15.1728	-15.1962	-15.2010	-15.1962
28	-33.521	-33.4993	-33.4998	-33.4871	-33.4998	-33.4816	-33.4998
29	31.692	31.6989	31.7019	31.6699	31.7019	32.6797	31.7066
30	43.582	43.6075	43.6008	43.4358	43.5999	43.9799	43.6026
31	44.188	44.1992	44.1977	43.7535	44.1971	43.7826	44.1971
32	-45.710	-45.7500	-45.7209	-45.8773	-45.7209	-45.9124	-45.7209
33	-36.572	-36.5825	-36.5808	-36.5657	-36.5807	-36.2987	-36.5803
34	-12.184	-12.1969	-12.1970	-12.1622	-12.1969	-11.6651	-12.1916
35	0.0723	0.0722	0.0723	0.0722	0.0723	0.0714	0.0723
36	0.0927	0.0925	0.0926	0.0897	0.0926	0.0886	0.0926
37	-0.0229	-0.0229	-0.0229	-0.0225	-0.0229	-0.0168	-0.0229
38	-0.0519	-0.0523	-0.0522	-0.0554	-0.0523	-0.0514	-0.0522
39	-0.0391	-0.0392	-0.0390	-0.0393	-0.0390	-0.0411	-0.0390
40	0.02538	0.0261	0.0254	0.0252	0.0254	0.0243	0.0254
41	0.0614	0.0614	0.0614	0.0616	0.0614	0.0635	0.0614
42	0.1061	0.1063	0.1063	0.1067	0.1063	0.1039	0.1063

Table 1: 34-node-system state estimates (73 equations; redundancy ratio=1.74)

1-34: nodal heads (m Aq) at nodes 1-34;

35-42: fixed-head nodes in/out flows (m^3/s) at nodes 27-34

LS - least squares method; LAV - least absolute values method

State Variable	Exact value	LS	LAV	LS (Ex.1)	LAV (Ex.1)	LS (Ex.2)	LAV (Ex.2)
1	32.638	32.6617	32.6612	32.6066	32.6627	33.6971	32.6729
2	43.749	43.7502	43.7556	43.3259	43.7558	43.8230	43.7582
3	46.041	46.0686	46.0521	45.2732	46.0523	45.8108	46.0727
4	46.618	46.6560	46.6375	45.8594	46.6348	46.4677	46.6616
5	43.265	43.2504	43.2630	43.0960	43.2631	43.8204	43.2691
6	43.024	42.9711	42.9758	43.0681	42.9776	44.1504	43.0001
7	42.402	42.3626	42.3805	42.8532	42.3911	43.8289	42.3930
8	42.130	42.0945	42.1063	42.8536	42.1301	43.9074	42.1309
9	43.798	43.7888	43.7864	43.4703	43.7799	44.6383	43.8150
10	47.950	47.9426	47.9439	47.7249	47.9465	48.0481	47.9471
11	44.664	44.6815	44.6720	44.1652	44.6703	45.0104	44.6708
12	44.004	44.0217	44.0158	43.6192	44.0067	44.6583	44.0395
13	49.274	49.3056	49.3031	49.0158	49.3063	49.7032	49.3149
14	49.099	49.0981	49.0952	48.9325	49.0959	49.2708	49.0975
15	49.057	49.0548	49.0519	48.8973	49.0505	49.2145	49.0505
16	49.298	49.3271	49.3242	49.0225	49.3272	49.7337	49.3366
17	47.970	47.9675	47.9673	47.7192	47.9694	48.1478	47.9702
18	49.338	49.3488	49.3456	49.0293	49.3485	49.7666	49.3585
19	49.029	49.0489	49.0437	48.5915	49.0457	49.2476	49.0582
20	46,618	46.6559	46.6375	45.8634	46.6348	46.4839	46.6615
21	45.623	45.6517	45.6397	45.0280	45.6367	45.8039	45.6635
22	46.588	46.6238	46.6075	45.8632	46.6059	46.4995	46.6316
23	48.379	48.3959	48.3877	47.8428	48.3893	48.4878	48.4053
24	43.249	43.2248	43.2267	43.1154	43.2230	44.2447	43.2529
25	42.532	42.4872	42.4966	42.9333	42.5095	44.0054	42.5202
26	32.086	32.1026	32.1014	32.0160	32.1046	33.2302	32.1154
27	-15.233	-15.1972	-15.1969	-15.1454	-15.1970	-15.2306	-15.1979
28	-33.521	-33.4996	-33.4993	-33.4792	-33.4998	-33.4738	-33.4999
29	31.692	31.7042	31.7025	31.5704	31.7061	32.7783	31.7168
30	43.582	43.6064	43.6031	43.3150	43.5904	44.3922	43.6294
31	44.188	44.2061	44.1981	43.5266	44.2015	43.9097	44.2011
32	-45.710	-45.7475	-45.7246	-45.9293	-45.7209	-45.9514	-45.7209
33	-36.572	-36.5821	-36.5807	-36.5597	-36.5807	-36.2977	-36.5801
34	-12.184	-12.1963	-12.1927	-12.1452	-12.1983	-11.6589	-12.1909
35	0.0723	0.0722	0.0723	0.0719	0.0723	0.0716	0.0723
36	0.0927	0.0925	0.0925	0.0886	0.0926	0.0874	0.0927
37	-0.0229	-0.0229	-0.0229	-0.0223	-0.0229	-0.0170	-0.0229
38	-0.0519	-0.0523	-0.0523	-0.0554	-0.0524	-0.0510	-0.0522
39	-0.0391	-0.0392	-0.0391	-0.0394	-0.0391	-0.0427	-0.0393
40	0.0254	0.0261	0.0257	0.0249	0.0254	0.0235	0.0254
41	0.0614	0.0614	0.0614	0.0619	0.0614	0.0633	0.0614
42	0.1061	0.1063	0.1063	0.1073	0.1063	0.1033	0.1062

Table 2: 34-node-system state estimates (59 equations; redundancy ratio 1.4)

1-34: nodal heads (m Aq) at nodes 1-34;

35-42: fixed-head nodes in/out flows (m^3/s) at nodes 27-34

LS - least squares method; LAV - least absolute values method